Student Information

Full Name : ANSWER SHEET
Id Number : ANSWER SHEET

Answer 1

a)

$G_R = (V_R, E_R)$ is a **digraph** to represent R. Where

$V_R = \{\emptyset, \{0\}, \{1\}, \{2\}, \{0,1\}, \{1,2\}, \{0,2\}, \{0,1,2\}\}$ and

$E_R = \{(\emptyset, \emptyset), (\emptyset, \{0\}), (\emptyset, \{1\}), (\emptyset, \{2\}), (\emptyset, \{0,1\}), (\emptyset, \{1,2\}), (\emptyset, \{0,2\}), (\emptyset, \{0,1,2\}), (\{0\}, \{0\}), (\{0\}, \{0,1\}), (\{0\}, \{0,2\}), (\{0\}, \{0,1,2\}), (\{1\}, \{1\}), (\{1\}, \{0,1\}), (\{1\}, \{1,2\}), (\{1\}, \{0,1,2\}), (\{2\}, \{2\}), (\{2\}, \{1,2\}), (\{2\}, \{0,2\}), (\{2\}, \{0,1,2\}), (\{0,1\}, \{0,1\}), (\{0,1\}, \{0,1,2\}), (\{1,2\}, \{1,2\}), (\{1,2\}, \{0,1,2\}), (\{0,2\}, \{0,2\}), (\{0,2\}, \{0,1,2\}), (\{0,1,2\}, \{0,1,2\})\}$

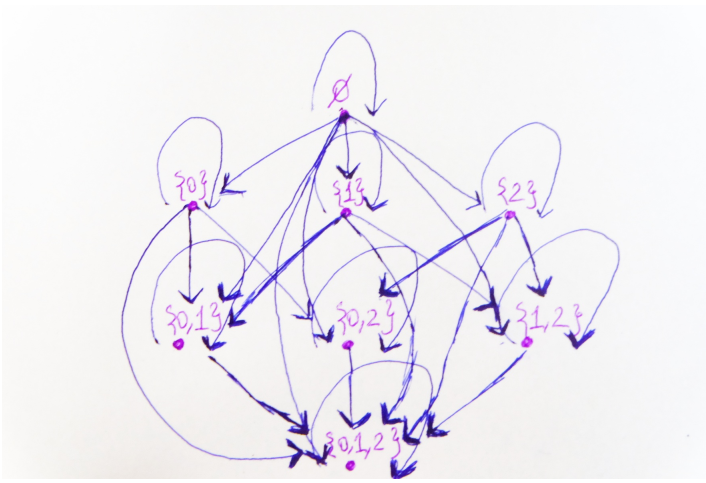


Figure 1: Directed graph representation of R. (G_R)

b)

(S,R) is a poset iff R on S is a *partial order relation* which requires R to be **reflexive**, **antisymmetric** and **transitive**.

R is **reflexive** since for all $x \in S$ $(x,x) \in R$ as x is a subset of itself.

R is **antisymmetric** for all $x, y \in S$ if $(x,y) \in R$ and $(y,x) \in R$ then $x = y$. Using G_R in a), you may deduce that R antisymmetric since apart from the self loops due to reflexivity, no two edges of the form (v_1, v_2) and (v_2, v_1) such that $v_1, v_2 \in R$ occur in E_R at the same time.

R is **transitive** since for all $x, y, z \in S$ if $(x,y) \in R$ and $(y,z) \in R$, it means that x is a subset of y and y is a subset of z then by the definition of subset x is must also be a subset of z.

c)

A totally ordered set is a partially ordered set in which every two elements are *comparable.* If we order the subsets of the set $\{0, 1, 2\}$ by inclusion (the boolean lattice on a set of size 3), **we don't get a total order** because $\{0, 1\}$ and $\{2\}$ are incomparable (there are no inclusion relations between them).

d)

Hasse diagram for (S,R): eliminate self-loops, directed arcs due to transitivity, and direction on arcs via introducing an order for undirected arcs (bottom-up). Hasse diagram is shown in the Figure-2.

$\emptyset$ is the **minimal element** and $\{0, 1, 2\}$ is the **maximal element**.

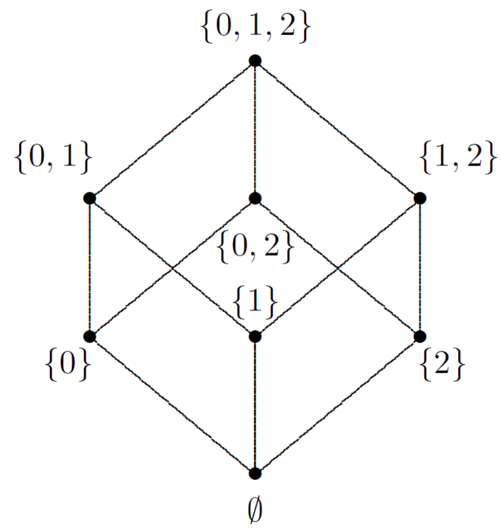


Figure 2: Hasse diagram for (S,R)

e)

The poset consisting of all the subsets of $\{0, 1, 2\}$ **is a lattice.** because for every pair of objects exists there is a unique *greatest lower bound* and *least upper bound.*
The greatest lower bound of two subsets is the intersection of the two subsets:
for example, $\{0, 1\} \wedge \{1, 2\} = \{0, 1\} \cap \{1, 2\} = \{1\}$.
The least upper bound is the union of the two subsets:
for example, $\{0, 1\} \vee \{1, 2\} = \{0, 1\} \cup \{1, 2\} = \{0, 1, 2\}$.

Answer 2

a)

vertex	adjacent vertices
a	-
b	a,c
c	f
d	a,c,d,e,g
e	c,f,g
f	b
g	d

b)

$$M_G = \begin{array}{c} \\ a \\ b \\ c \\ d \\ e \\ f \\ g \end{array} \begin{array}{c} \begin{array}{ccccccc} a & b & c & d & e & f & g \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

c)

vertex,v	$\deg^+(v)$	$\deg^-(v)$
a	2	0
b	1	2
c	3	1
d	2	5
e	1	3
f	2	1
g	2	1

d)

$e - g - d - f$
$e - c - f - b$
$e - f - b - a$
$d - e - f - b$
$g - d - e - c$
$g - d - e - f$

e)

Simple Circuit: A circuit in which the only repeated vertices are the first and last vertices.
$d - e - g - d$
$g - d - e - g$
$e - g - d - e$
$b - c - f - b$
$f - b - c - f$
$c - f - b - c$

f)

Oriented graph $G = (V, E)$ is weakly connected graph if and only if for every two vertices $u, v \in V$ exists a directed path from u to v or directed path from v to u.

The underlying graph of G is G' and it is provided in Figure-3. G' is connected because there is a path between every pair of vertex.

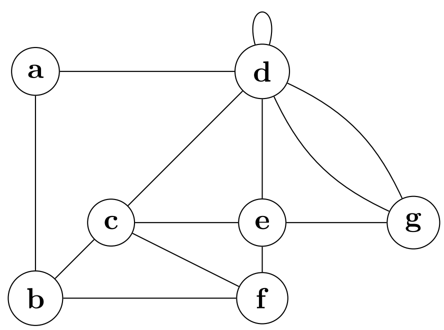


Figure 3: Graph G' (Grapn in Q2 undirected representation).

When you perform DFS on G' it yields to the tree in Figure-4. And every tree is connected by definition. **Hence G is weakly-connected.**

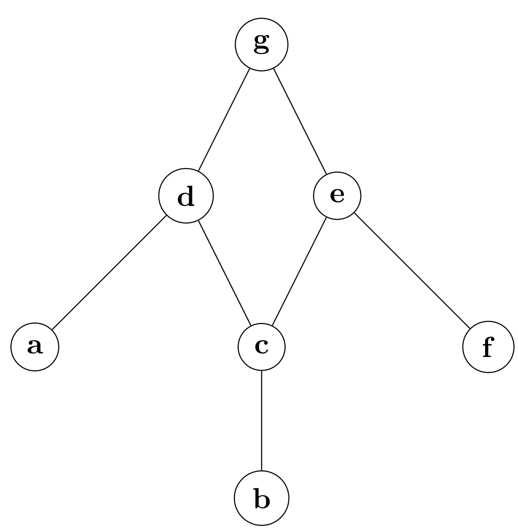


Figure 4: The tree created after by applying DFS to the undirected graph provided in the previous picture.

g)

A directed graph is strongly connected if there is a path between all pairs of vertices. A strongly connected component (SCC) of a directed graph is a maximal strongly connected subgraph.

SCC's of G are: $\{b, c, f\}, \{a\}, \{d, e, g\}$

h)

Let G be a graph with adjacency matrix A with respect to the ordering $v_1, v_2, \cdots v_n$ of the vertices of the graph (with directed or undirected edges, with multiple edges and loops allowed). The number of different paths of length r from v_i to v_j , where r is a positive integer, equals the $(i,j)_{th}$ entry of A^r.

Step 1: Find adjacency matrix of H:

$$\text{H}= \begin{array}{c} \\ \text{d} \\ \text{e} \\ \text{f} \\ \text{g} \end{array} \begin{array}{c} \begin{array}{cccc} \text{d} & \text{e} & \text{f} & \text{g} \end{array} \\ \begin{bmatrix} 1 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix} \end{array}$$

Step 2: Find H^2:

$$H^2= \begin{array}{c} \\ \text{d} \\ \text{e} \\ \text{f} \\ \text{g} \end{array} \begin{array}{c} \begin{array}{cccc} \text{d} & \text{e} & \text{f} & \text{g} \end{array} \\ \begin{bmatrix} 2 & 1 & 1 & 2 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 \end{bmatrix} \end{array}$$

Step 3: Find H^3 and find the length value from d to g which is 3.

$$H^3= \begin{array}{c} \\ \text{d} \\ \text{e} \\ \text{f} \\ \text{g} \end{array} \begin{array}{c} \begin{array}{cccc} \text{d} & \text{e} & \text{f} & \text{g} \end{array} \\ \begin{bmatrix} 4 & 2 & 1 & \boldsymbol{3} \\ 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 2 & 1 & 1 & 2 \end{bmatrix} \end{array}$$

Notice that we can use non-simple paths. The three paths found in matrix calculation are:

$d - d - d - g$

$d - d - e - g$

$d - g - d - g$

Answer 3

a)

If a graph G has an Euler path, then it must have exactly two odd vertices. The degrees of the vertices are:

$a : 2$, $b : 3$, $c : 2$, $d : 5$, $e : 4$, $f : 2$, $g : 2$, $h : 2$ (b and d are odd vertices).

Then the graph has an Euler path.

b)

The graph has an Euler path but has no Euler circuit. It has two odd vertices B and D. Euler theorems say if a graph has odd vertices, then the graph has no Euler circuit; if the graph is connected and has only two odd vertices, then the graph has an Euler path.

c)

Yes it has a hamiltonian path f-g-h-e-d-a-b-c.

d)

The graph has no Hamilton circuit. If one travels from A, B, C, D to E, F, G, H and comes back, D and E will be visited at least twice. Such a circuit is not a Hamilton circuit.

Answer 4

The graphical arrangement of the vertices and edges makes them look different but nevertheless, they are the same graph.
Formally:
The simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ are isomorphic if there is a bijective function f from V_1 to V_2 with the property that a and b are adjacent in G_1 if and only if $f(a)$ and $f(b)$ are adjacent in G_2 $\forall a, b \in V_1$.

Let f be a bijective function from V to V'. Let the correspondence between the graphs be:
$a' = f(a)$
$b' = f(e)$
$c' = f(c)$
$d' = f(d)$
$e' = f(b)$
The above correspondence preserves adjacency as a is adjacent to b and e in G, and $f(a) = a'$ is adjacent to $f(b) = e'$ and $f(e) = b'$ in G'.
Similarly, it can be shown that the adjacency is preserved for all vertices. Hence, G and G' are isomorphic.

Answer 5

a)

step 1: Initialization
Set the l() values of the neighbors of a to the length of the edge between a and them. Mark a with a $(*)$ since the shortest path to itself is found whose length is 0. For the other vertices set the l() value to ∞.
step 2:
The next vertex is b which has the minimum distance of 3. We update l() values of c and f.
step 3:
The next one is h, f and i are updated.
step 4:
Arbitrarily chose c as the new one. f, g and d are updated.
step 5:
The new one is e. No update.
step 6:
The next is i. j is updated.
step 7:

The next is f. j is updated.
step 8:
The next is d. k is updated.
step 9:
Randomly choose j. **Path is** a, b, c, f, j **and the length is** 10.

	new	a	b	c	d	e	f	g	h	i	j	k
st1	a	*0	3	∞	∞	5	∞	∞	4	∞	∞	∞
st2	b	*0	*3	5	∞	5	10	∞	4	∞	∞	∞
st3	h	*0	*3	5	∞	5	9	∞	*4	6	∞	∞
st4	c	*0	*3	*5	8	5	7	11	*4	6	∞	∞
st5	e	*0	*3	*5	8	*5	7	11	*4	6	∞	∞
st6	i	*0	*3	*5	8	*5	7	11	*4	*6	12	∞
st7	f	*0	*3	*5	8	*5	*7	11	*4	*6	10	∞
st8	d	*0	*3	*5	*8	*5	*7	11	*4	*6	10	10
st9	j	*0	*3	*5	*8	*5	*7	11	*4	*6	***10**	10

b)

We will create a list of visited nodes, starting from a:

visited = { a }

Using Prim's algorithm, the smallest weighted adjacent edge is (a, b, 3).

visited = { a, b }

Then, the least weighted edge from the list is to c with edge (b, c, 2).

visited = { a, b, c }

Then, the least weighted edge from the list is to vertex f with edge (c, f, 2).

visited = { a, b, c, f }

Then, the least weighted edge from the list is to vertex d with edge (c, d, 3).

visited = { a, b, c, f, d }

Then, the least weighted edge from the list is to vertex k with edge (d, k, 2).

visited = { a, b, c, f, d, k }

Then, the least weighted edge from the list is to vertex j with edge (f, j, 3).

visited = { a, b, c, f, d, k, j }

Then, the least weighted edge from the list is to vertex e with edge (f, e, 4).

visited = { a, b, c, f, d, k, j, e }

Then, the least weighted edge from the list is to vertex i with edge (f, i, 4).

visited = { a, b, c, f, d, k, j, e, i }

Then, the least weighted edge from the list is to vertex h with edge (i, h, 2).

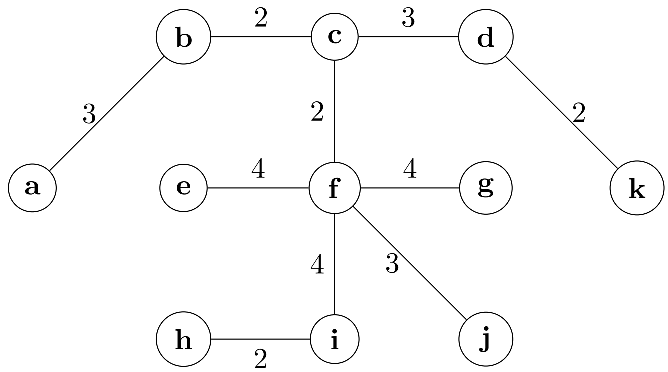
visited = { a, b, c, f, d, k, j, e, i, h }

Then, the final least weighted edge from the list is to vertex g with edge (f, g, 4).

$$\text{visited} = \{ \text{a, b, c, f, d, k, j, e, i, h, g} \}$$

Thus, all the nodes has been visited and Prim's algorithm has been complete. The resulting spanning tree, with a total weight of 29, is the following:

Figure 5: Minimum Spanning Tree of Graph G



Answer 6

a)

There are 7 vertices and 6 edges (Always $|v|$ vertices and $|v-1|$ edges).
The height is 3, it can be reached by the paths A, C, E, G or A, C, E, F.

b)

$a - b - c - d - e - f - g$

c)

$b - d - f - g - e - c - a$

d)

$b - a - d - c - f - e - g$

e)

A binary tree T is full if each node is either a leaf or possesses exactly two child nodes. In this sense our tree T **is a full binary tree**.

f)

A binary tree T with n levels is complete if all levels except possibly the last are completely full, and the last level has all its nodes to the left side. Our tree T **is not complete because not all the levels are full**.

g)

No it is not because $f : 23 > c : 24$ should hold, but it does not.

h)

A height 5 full binary tree with minimum possible vertices is drawn below. There are 11 vertices.

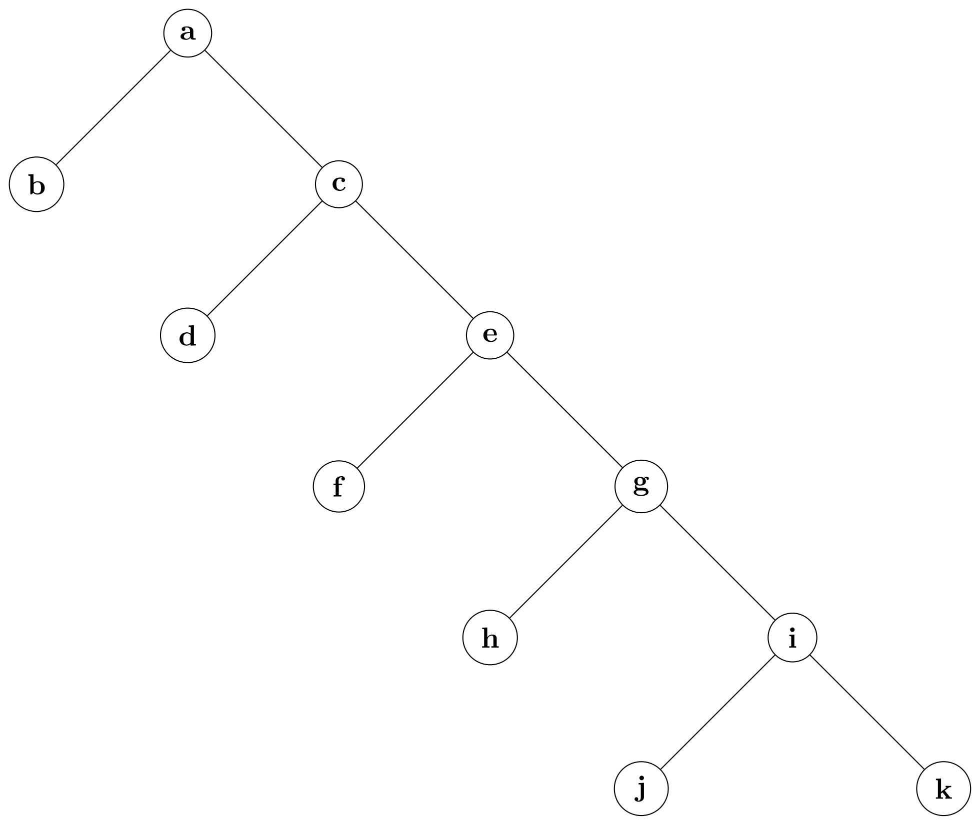


Figure 6: A minimum vertice full binary tree of height 5.

i)

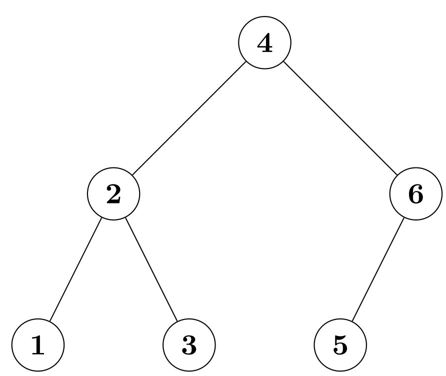


Figure 7: Complete tree with set of integer keys 1,2,3,4,5,6.

j)

$4 - 2 - 1$
$4 - 6$

k)

A minimum spanning tree for the tree in $Q2$.

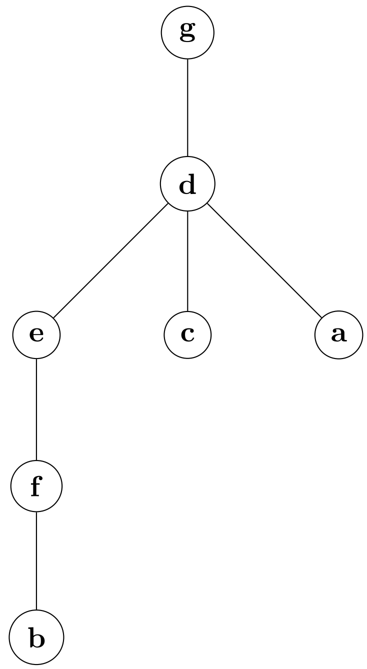


Figure 8: A minimum spanning tree for the tree in $Q2$

l)

A BST with max height should have a linear structure which would yield a height of $k - 1$.